Mathematics

HP COMPUTER CURRICULUM

Functions

TEACHERS ADVISOR

HEWLETT PACKARD

Hewlett-Packard
Computer Curriculum Series

mathematics
TEACHER'S ADVISOR

functions

by E. A. Herber
Golden High School
Golden, Colorado

edited by Jean Danver and Christine Doerr
Hewlett-Packard

Hewlett-Packard Company
11000 Wolfe Road
Cupertino, California 95014

HP 5951-4446

October 1972

The Hewlett-Packard Computer Curriculum Series represents the published results of a Curriculum Development project sponsored by the Data Systems Division of Hewlett-Packard Company. This project is under the directorship of Jean H. Danver.

This material is designed to be used with any Hewlett-Packard system with the BASIC programming language such as the 9830A Educational BASIC, and the 2000 and 3000 series systems.

The Hewlett-Packard Company makes no warranty expressed or implied, and assumes no responsibility in connection with the operation of the program material contained herein.

TABLE OF CONTENTS

INTRODUCTION

This Mathematics Set of the Hewlett-Packard Computer Curriculum Series consists of a Student Lab Book and a corresponding Teacher's Advisor. It was designed to help meet the need for computer-oriented problems in mathematics, providing students an opportunity to use a computer as a problem solving tool within a particular subject matter area.

The materials are designed for flexible use as desired by the individual instructor. The material and exercises in this unit are intended as an "enrichment" experience in the field of functions. The unit begins with a standard definition of a function. Then, in the first of two major sections, finite functions are used to illustrate such properties as the inverse of a function, increasing and decreasing functions, and symmetry. In the second section, infinite functions are used to investigate other properties such as the zeros of a function, continuity, and relative maximums and minimums. The unit is not intended to be a complete study of functions; instead, each discussion is self-contained, so the concepts can be studied independently and in any sequence. Thus, the material can be used to supplement and enrich your curriculum in any fashion you choose.

The mathematical concepts needed for each exercise are briefly reviewed, but you may want your students to study these in greater detail before attempting the exercises, especially if their background in functions is limited. A list of possible references is included at the end of each of the two major sections. These references will also provide additional problems for your better students.

The degree of difficulty of the material is dependent upon the amount of assistance which you choose to provide. With no assistance, the better student should be challenged. However, given a good deal of assistance, any second year algebra student should be able to work out the exercises with no great difficulty. The level of the material is determined by the assumption that students taking second year algebra will be quite capable as a group.

The Lab Book provides text material and programming exercises for the students. There is a problem analysis, including a suggested approach and a macro flow chart, for each exercise. The Teacher's Advisor contains an example of a program to solve each exercise, micro flow charts, and a brief discussion of the important elements of the exercise. The micro flow charts should be given to the students only after they have made an attempt to solve the problem on their own. For best results, you should study all the solutions until you are certain you have a complete grasp of the general methods. This should be done before assigning the material to the class.

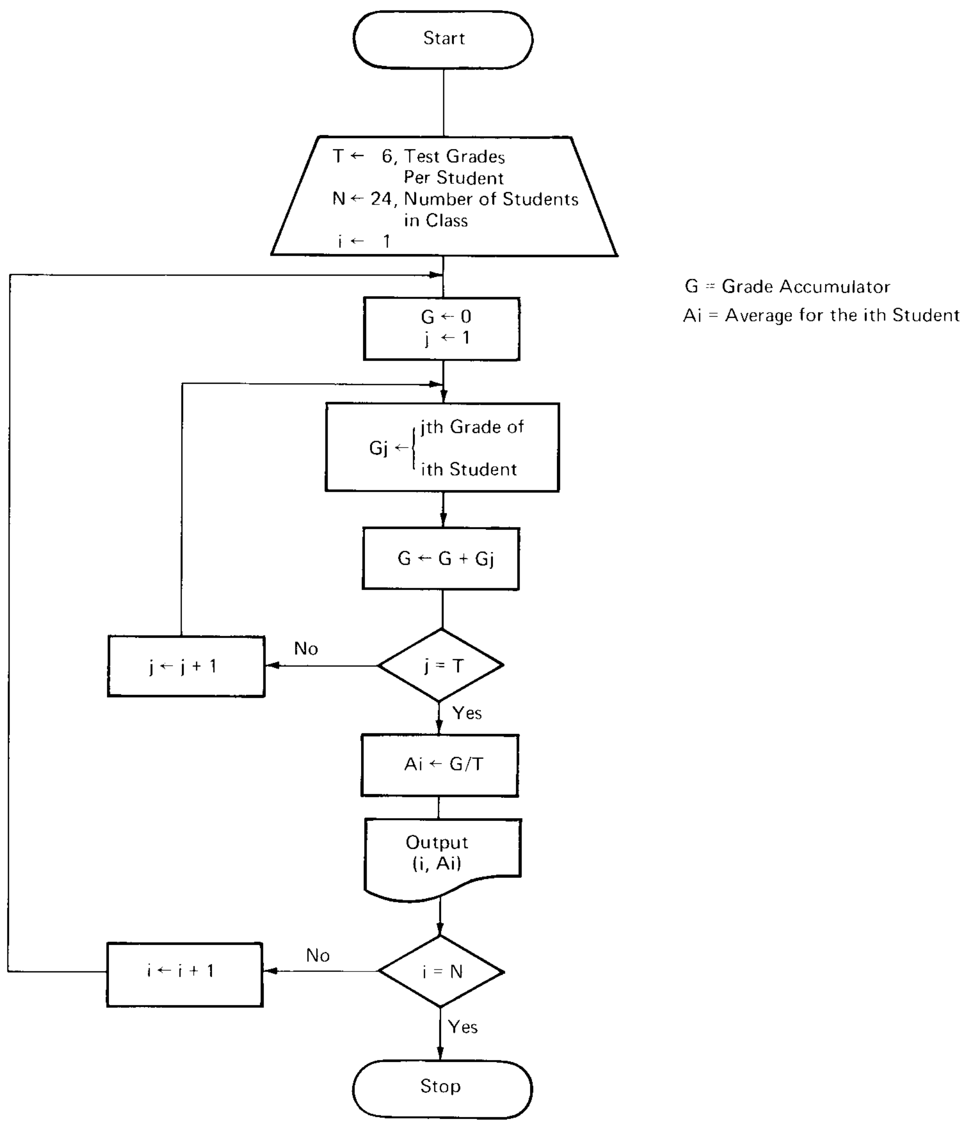
You will undoubtedly think of different programming methods or techniques as you study the example programs. Encourage the students to do the same. There are no *approved* solutions. All solutions are acceptable if they produce the correct results. At this level, there is no need for emphasis on the efficiency of a student's program. The important question is, does it work?

DEFINITION OF A FUNCTION

Exercise 1 — Averaging Grades as an Example of a Function

(a) A program to compute the grade average for a class of 24 students for a 9 week grading period. The output is printed in ordered pair form.

Micro Flow Chart
Exercise 1(a)



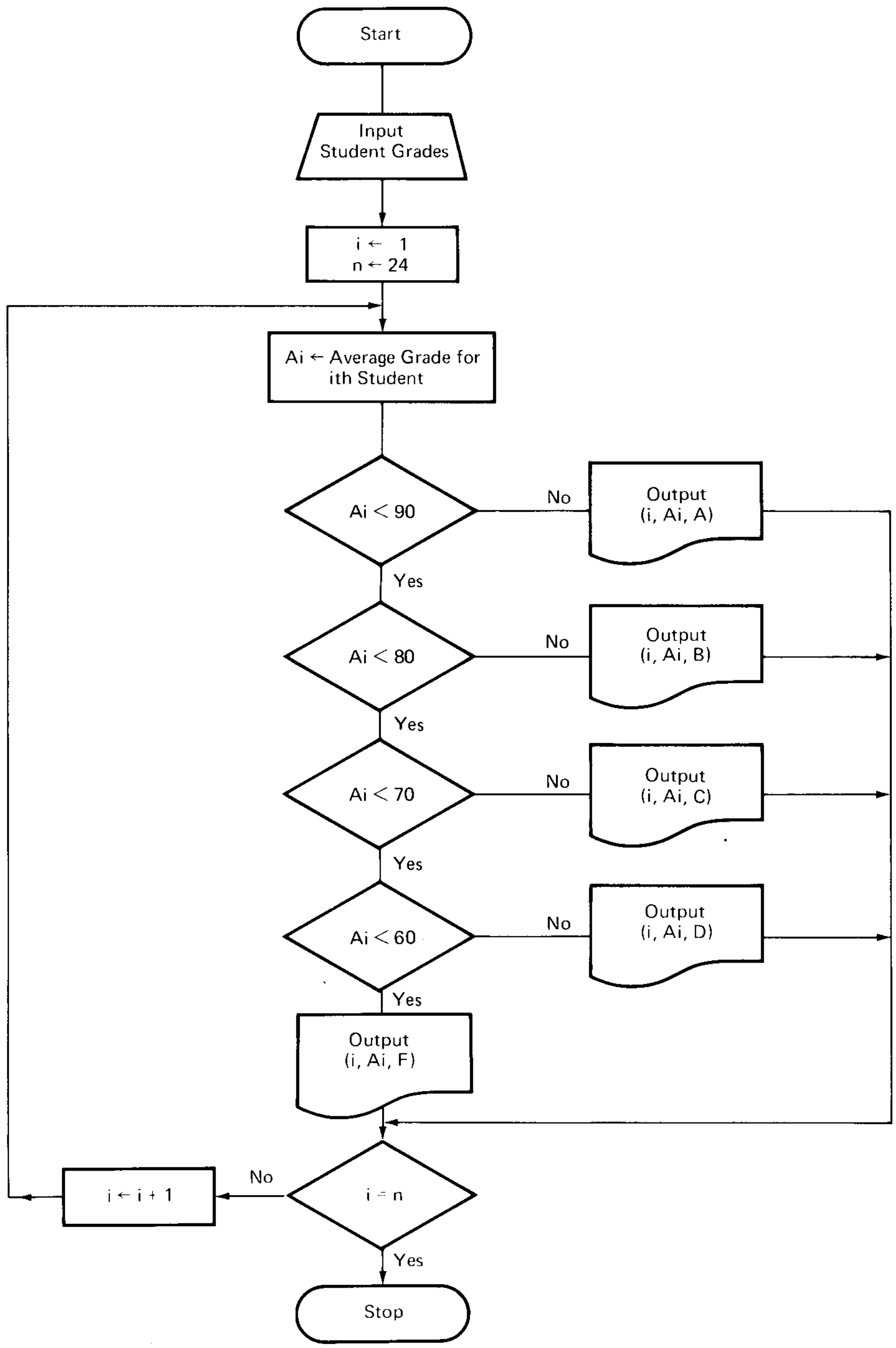
Example Program

Exercise 1(a)

```
10   REM--THIS PROGRAM WILL COMPUTE THE AVERAGE GRADE FOR A CLASS
11   REM--OF N STUDENTS BASED ON T GRADES FOR ONE GRADING PERIOD.
12   REM--THE OUTPUT IS IN ORDERED PAIR (FUNCTION) FORM.  ENTER
13   REM--ON THE DATA LINE N,T (NUMBER OF STUDENTS,NUMBER OF TESTS) AND
14   REM--THE TEST SCORES OF EACH STUDENT, NO.1 THROUGH NO. N.
15   DATA 24,6,100,90,98,72,96,88,100,95,90,100,98,100,60,80,62,63,76,68
16   DATA 100,70,84,90,80,80,100,90,92,70,88,78,85,65,84,70,84,74
17   DATA 100,100,94,92,94,100,100,95,90,87,96,100,100,70,73,80,88,80
18   DATA 100,75,82,87,84,88,100,90,88,100,94,96,70,75,81,80,84,80
19   DATA 100,65,88,72,80,80,85,90,88,87,86,84,80,70,46,55,70,64,75,70,58
20   DATA 57,70,56,80,65,66,66,74,60,100,78,73,75,60,96,95,55,61,60,60
21   DATA 100,90,61,57,58,60,80,85,85,70,80,64,80,80,80,40,68,84
22   DATA 100,50,84,75,84,76,85,50,54,68,50,85,60,56,55,74,68
25   DIM A[200]
30   READ N,T
50   FOR I=1 TO N
55   LET G=0
60   FOR J=1 TO T
70   READ G[J]
80   LET G=G+G[J]
90   NEXT J
100   LET A[I]=G/T
110   PRINT "(STUDENT NO.";I;",";A[I];")"
120   NEXT I
130   END
```

```
RUN

(STUDENT NO. 1      , 90.6667      )
(STUDENT NO. 2      , 97.1667      )
(STUDENT NO. 3      , 68.1667      )
(STUDENT NO. 4      , 84    )
(STUDENT NO. 5      , 86.3333      )
(STUDENT NO. 6      , 77    )
(STUDENT NO. 7      , 96.6667      )
(STUDENT NO. 8      , 94.6667      )
(STUDENT NO. 9      , 81.8333      )
(STUDENT NO. 10     , 86    )
(STUDENT NO. 11     , 94.6667      )
(STUDENT NO. 12     , 78.3333      )
(STUDENT NO. 13     , 80.8333      )
(STUDENT NO. 14     , 86.6667      )
(STUDENT NO. 15     , 64.1667      )
(STUDENT NO. 16     , 64.3333      )
(STUDENT NO. 17     , 68.5         )
(STUDENT NO. 18     , 80.3333      )
(STUDENT NO. 19     , 71.8333      )
(STUDENT NO. 20     , 67.6667      )
(STUDENT NO. 21     , 77.3333      )
(STUDENT NO. 22     , 75.3333      )
(STUDENT NO. 23     , 75.6667      )
(STUDENT NO. 24     , 61.1667      )

DONE
```

Micro Flow Chart
Exercise 1(b)



Example Program

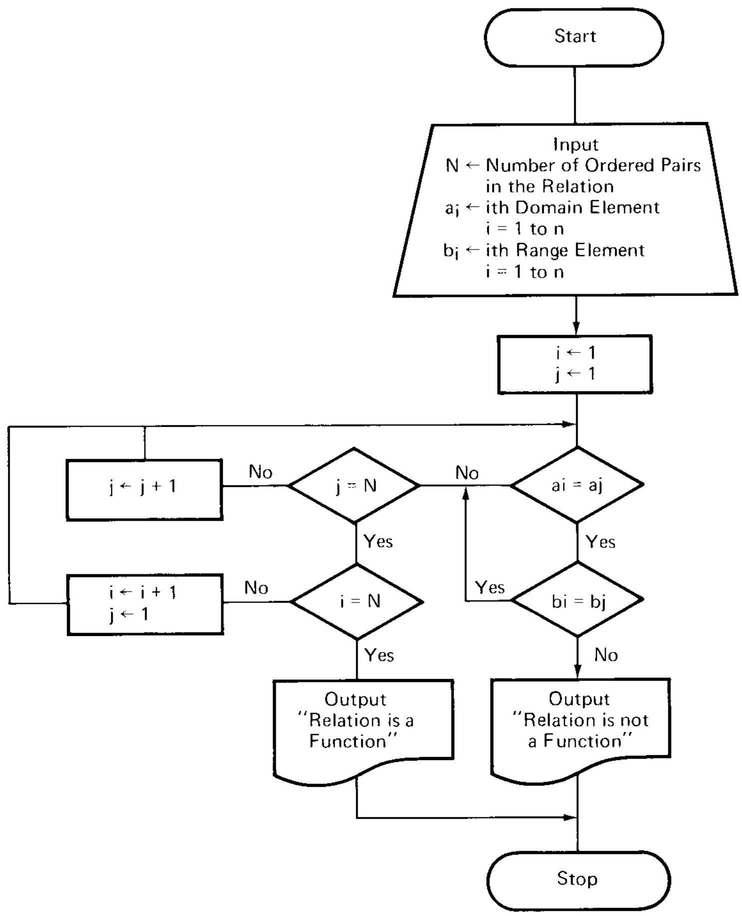
Exercise 1(b)

```
LIST

10  REM--THIS PROGRAM WILL COMPUTE THE AVERAGE GRADE FOR A CLASS
20  REM--OF N STUDENTS BASED ON T GRADES FOR ONE GRADING PERIOD.
30  REM--THE OUTPUT IS IN ORDERED PAIR (FUNCTION) FORM.  ENTER
40  REM --ON THE DATA LINE N, T, AND THE TEST SCORES OF EACH
50  REM--STUDENT, NO.1 THROUGH STUNDENT NO. N.
60  DATA 24,6,100,90,98,72,96,88,100,95,90,100,98,100,60,80,62,63,76,68
70  DATA 100,70,84,90,80,80,100,90,92,70,88,78,85,65,84,70,84,74
80  DATA 100,100,94,92,94,100,100,95,90,87,96,100,100,70,73,80,88,80
90  DATA 100,75,82,87,84,88,100,90,88,100,94,96,70,75,81,80,84,80
100  DATA 100,65,88,72,80,80,85,90,88,87,86,84,80,70,46,55,70,64,75,70,8
110  DATA 57,70,56,80,65,66,66,74,60,100,78,73,75,60,96,95,55,61,60,60
120  DATA 100,90,61,57,58,60,80,85,85,70,80,64,80,80,80,40,68,84
130  DATA 100,50,84,75,84,76,85,50,54,68,50,85,60,56,55,74,68
140  DIM A[200]
150  READ N,T
160  FOR I=1 TO N
170  LET G=0
180  FOR J=1 TO T
190  READ G[J]
200  LET G=G+G[J]
210  NEXT J
220  LET A[I]=G/T
230  IF A[I]<90 THEN 260
240  PRINT "(STUDENT NO.";I;",";A[I];" A)"
250  GOTO 360
260  IF A[I]<80 THEN 290
270  PRINT "(STUDENT NO.";I;",";A[I];" B)"
280  GOTO 360
290  IF A[I]<70 THEN 320
300  PRINT "(STUDENT NO.";I;",";A[I];" C)"
310  GOTO 360
320  IF A[I]<60 THEN 350
330  PRINT "(STUDENT NO.";I;",";A[I];" D)"
340  GOTO 360
350  PRINT "(STUDENT NO.";I;",";A[I];" F)"
360  NEXT I
370  END
```

```
(STUDENT NO. 1    , 90.6667      A)
(STUDENT NO. 2    , 97.1667      A)
(STUDENT NO. 3    , 68.1667      D)
(STUDENT NO. 4    , 84    B)
(STUDENT NO. 5    , 86.3333      B)
(STUDENT NO. 6    , 77    C)
(STUDENT NO. 7    , 96.6667      A)
(STUDENT NO. 8    , 94.6667      A)
(STUDENT NO. 9    , 81.8333      B)
(STUDENT NO. 10   , 86    B)
(STUDENT NO. 11   , 94.6667      A)
(STUDENT NO. 12   , 78.3333      C)
(STUDENT NO. 13   , 80.8333      B)
(STUDENT NO. 14   , 86.6667      B)
(STUDENT NO. 15   , 64.1667      D)
(STUDENT NO. 16   , 64.3333      D)
(STUDENT NO. 17   , 68.5         D)
(STUDENT NO. 18   , 80.3333      B)
(STUDENT NO. 19   , 71.8333      C)
(STUDENT NO. 20   , 67.6667      D)
(STUDENT NO. 21   , 77.3333      C)
(STUDENT NO. 22   , 75.3333      C)
(STUDENT NO. 23   , 75.6667      C)
(STUDENT NO. 24   , 61.1667      D)

DONE
```

Exercise 2 — Modeling the Definition of a Function

Micro Flow Chart
Exercise 2



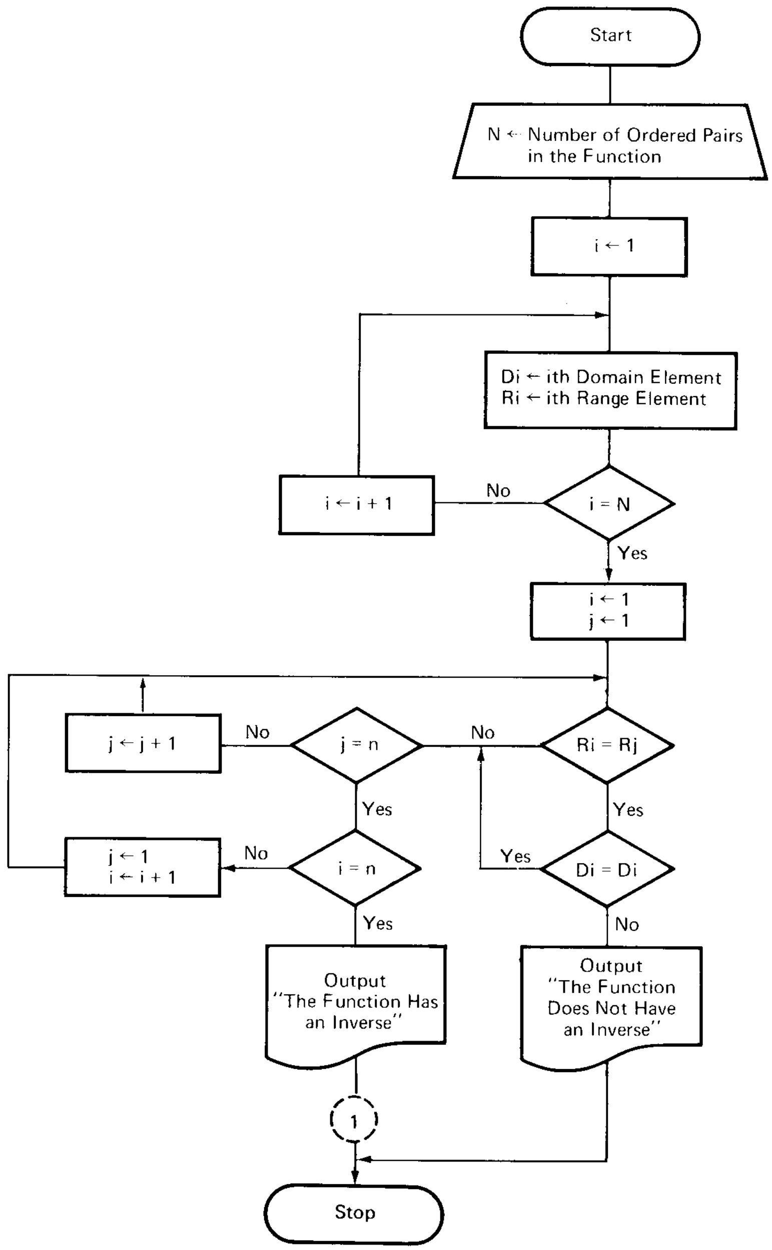
Example Program
Exercise 2

```
1  DATA 6,3,2,-6,8,3,1,8,16,-.5,-2,4,2,7,7,1,-8,6,14,2,.8,3,6,3,10
2  DATA 0,-8,6,5,2,5,-3,.75,17,.3,2,-.7,3,5
10  REM--THIS PROGRAM DETERMINES IF A GIVEN FINITE SET OF ORDERED
11  REM--PAIRS IS A FUNCTION.  ENTER THE NUMBER OF ORDERED PAIRS
12  REM--N, IN THE RELATION,  THEN THE ORDERED PAIRS PAIR BY PAIR.
30  READ N
40  FOR I=1 TO N
50  READ A[I],B[I]
60  NEXT I
100  FOR I=1 TO N
110  FOR J=1 TO N
120  IF A[I]=A[J] THEN 150
130  IF J=N THEN 180
140  NEXT J
150  IF B[I]=B[J] THEN 130
160  PRINT "RELATION IS NOT A FUNCTION"
170  GOTO 30
180  IF I=N THEN 200
190  NEXT I
200  PRINT "THE RELATIONS IS A FUNCTION"
205  GOTO 30
210  END

RUN


RELATION IS NOT A FUNCTION
THE RELATIONS IS A FUNCTION
RELATION IS NOT A FUNCTION

OUT OF DATA  IN LINE 30
```

USING INFINITE FUNCTIONS TO INVESTIGATE FUNCTION PROPERTIES

Exercise 3 — Finding the Inverse of a Function

Micro Flow Chart
Exercise 3(a)

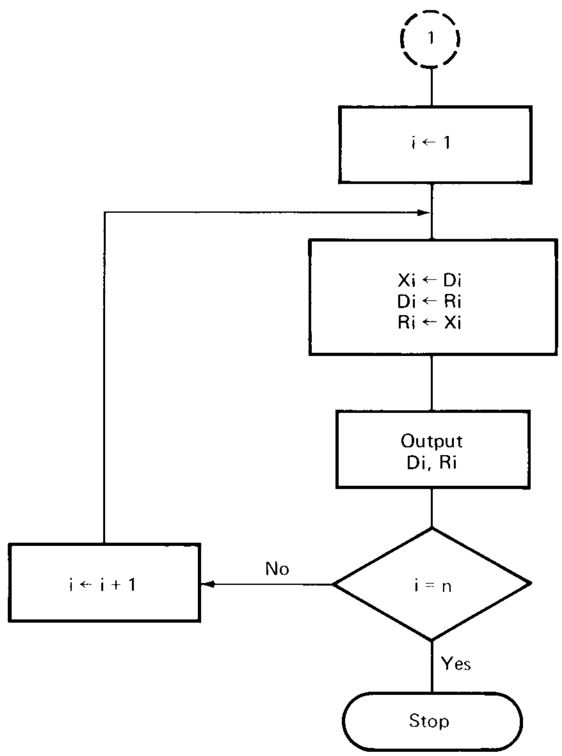


Example Program
Exercise 3(a)

```
10   DATA 5,-1,2,-3,5,7,-2,-3,5,.8,-.6
20   REM--THIS PROGRAM WILL DETERMINE IF A GIVEN FUNCTION DEFINED
30   REM--BY A FINITE SET OF ORDERED PAIRS HAS AN INVERSE.  ENTER
40   REM--ON THE DATA LINE THE NUMBER OF ORDERED PAIRS INTHE SET
50   REM--FOLLOWED BY THE ORDERQD PAIRS.
60   READ N
70   FOR I=1 TO N
80   READ D[I],R[I]
90   NEXT I
100  FOR I=1 TO N
110  FOR J=1 TO N
120  IF R[I]=R[J] THEN 160
130  NEXT J
140  NEXT I
150  GOTO 190
160  IF D[I]=D[J] THEN 130
170  PRINT "THE FUNCTION DOES NOT HAVE AN INVERSE"
180  GOTO 60
190  PRINT "THE FUNCTION HAS AN INVERSE"
200  GOTO 60
210  END

RUN


THE FUNCTION HAS AN INVERSE

OUT OF DATA  IN LINE 60
10 DATA 6,7,15,6,2,-7,3,2,1,0,15,-.5,10
RUN


THE FUNCTION DOES NOT HAVE AN INVERSE

OUT OF DATA  IN LINE 60
10 DATA 5,6,3,8,2,-1,6,-.8,.2,3,-7
RUN


THE FUNCTION HAS AN INVERSE

OUT OF DATA  IN LINE 60
```

(b) The student is asked to write a program that will cause an interchange of variable assignments of corresponding range elements. This is not necessary for an output of the inverse of a set of ordered pairs, but the exercise serves as an introduction to the interchange technique required in a later exercise.

Micro Flow Chart
Exercise 3(b)



Insert this section where indicated in the flow chart of (a) above

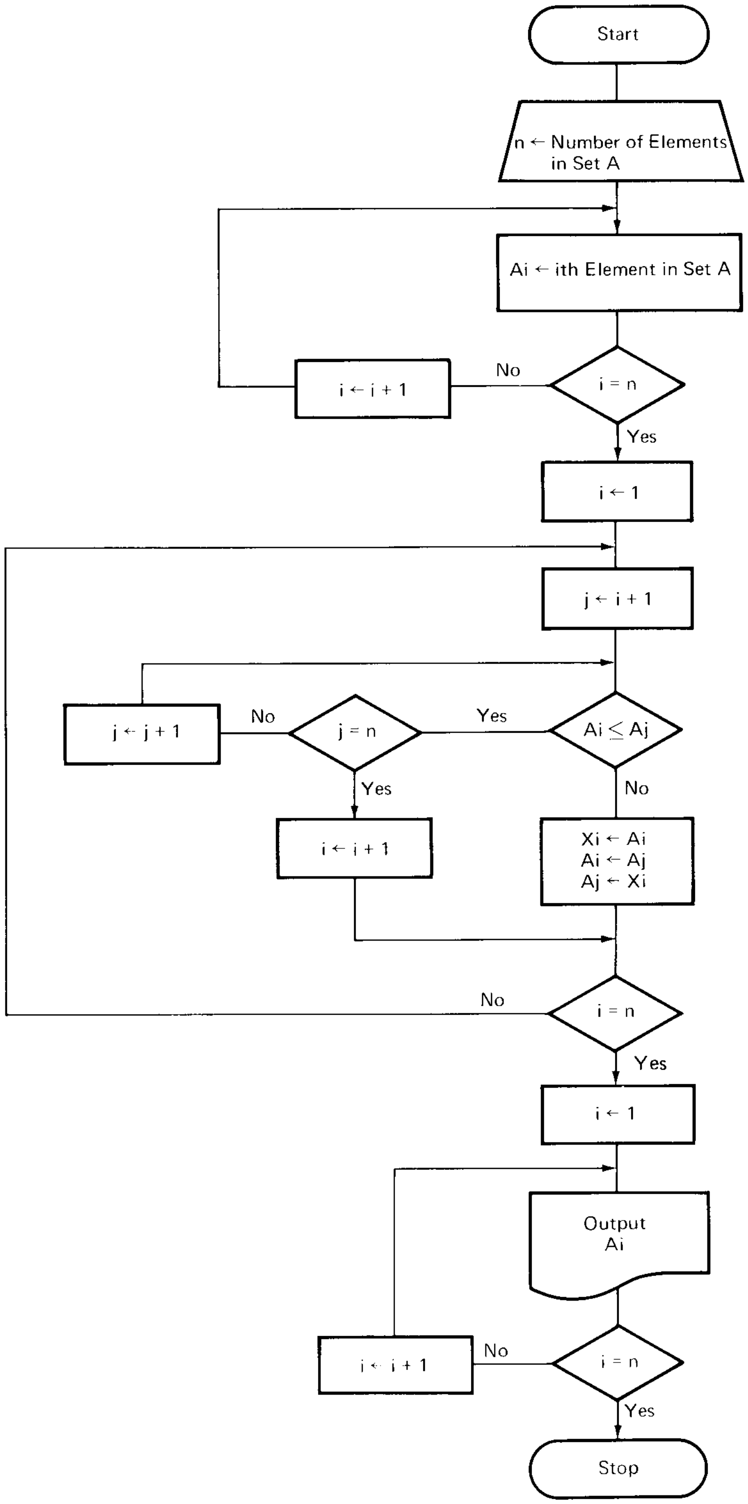
Example Program
Exercise 3(b)

```
10  DATA 5,6,3,8,2,-1,6,-.8,.2,3,-7,6,7,15,6,2,-7,3,2,1,0,15,-.5,10
20  DATA 5,-1,2,-3,5,7,-2,-3,5,.8,-.6
30  REM--THIS PROGRAM WILL PRINT THE INVERSE OF A GIVEN
40  REM--FUNCTION, IF IT EXISTS.
50  REM-- ENTER ON THE DATA LINE THE NUMBER OF
60  REM--ORDERED PAIRS IN THE SET, THE ORDERED PAIRS, PAIR
70  REM--BY PAIR.
80  READ N
90  FOR I=1 TO N
100  READ D[I],R[I]
110  NEXT I
120  FOR I=1 TO N
130  FOR J=1 TO N
140  IF R[I]=R[J] THEN 180
150  NEXT J
160  NEXT I
170  GOTO 220
180  IF D[I]=D[J] THEN 150
190  PRINT "THE FUNCTION DOES NOT HAVE AN INVERSE"
200  PRINT ""
210  GOTO 80
220  PRINT "THE FUNCTION HAS AN INVERSE"
230  PRINT ""
240  FOR I=1 TO N
250  LET X[I]=D[I]
260  LET D[I]=R[I]
270  LET R[I]=X[I]
280  PRINT "(";D[I];",";R[I];")"
290  NEXT I
300  PRINT ""
310  GOTO 80
320  END
```

```
RUN


THE FUNCTION HAS AN INVERSE

( 3      , 6      )
( 2      , 8      )
( 6      ,-1      )
( .2               ,-.8               )
(-7      , 3      )

THE FUNCTION DOES NOT HAVE AN INVERSE

THE FUNCTION HAS AN INVERSE

( 2      ,-1      )
( 5      ,-3      )
(-2      , 7      )
( 5      ,-3      )
(-.6               , .8               )


OUT OF DATA  IN LINE 80
```

Exercise 4 — Arranging Numbers in Ascending Order

Micro Flow Chart
Exercise 4

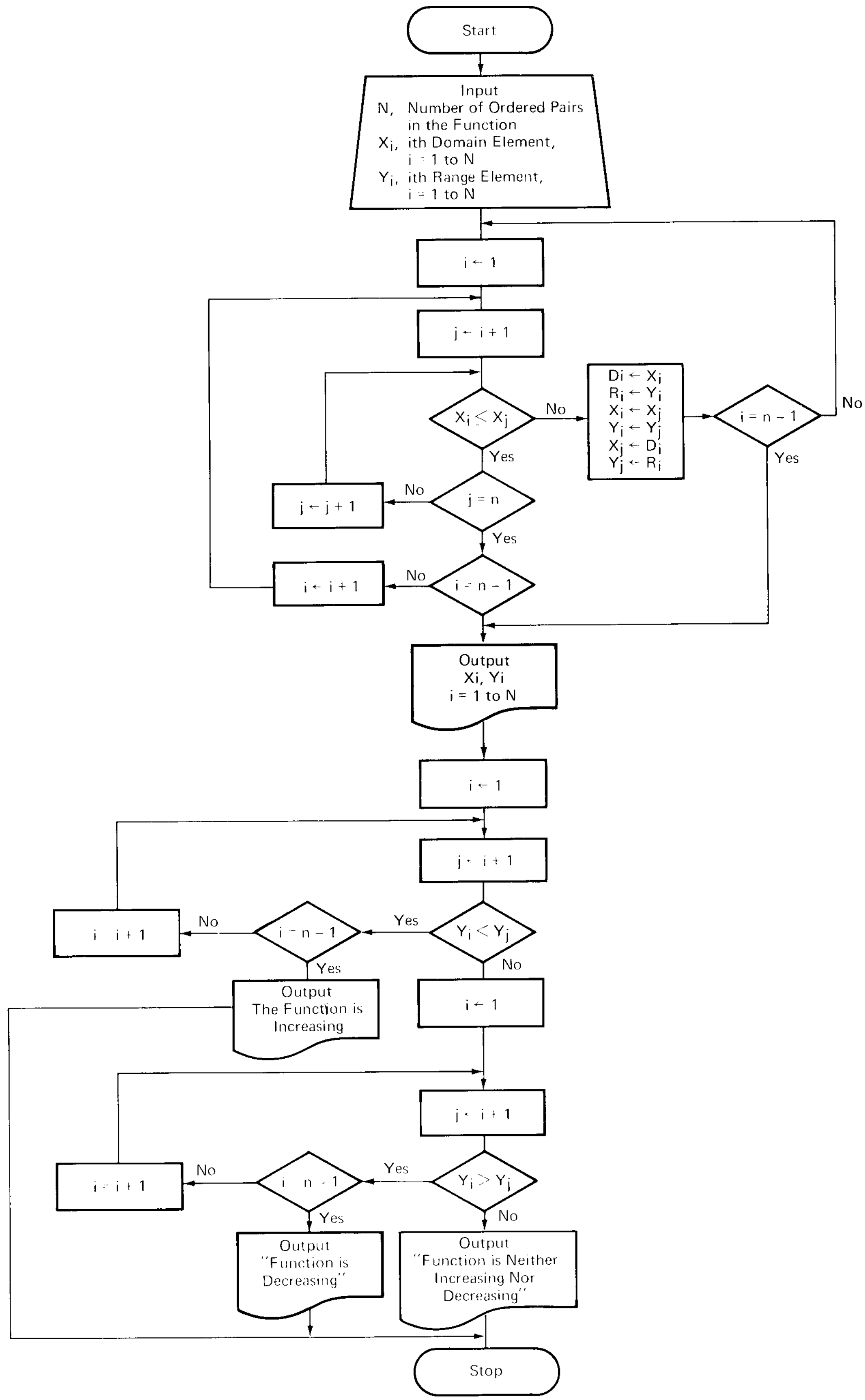


Example Program
Exercise 4

```
1   DATA 6,6,-2,8,5,0,-8
10  REM--THIS PROGRAM WILL ARRANGE A FINITE SET OF REAL NUMBERS
11  REM--IN ASCENDING ORDER.  ON THE DATA LINE ENTER THE NUMBER
12  REM--OF ELEMENTS IN THE SET AND THE ELEMENTS OF THE SET.
20  READ N
30  FOR I=1 TO N
40  READ A[I]
50  NEXT I
60  I=1
70  J=I+1
80  IF A[I] <= A[J] THEN 140
90  X[I]=A[I]
100  A[I]=A[J]
110  A[J]=X[I]
120  IF I=N THEN 170
130  GOTO 70
140  IF J=N THEN 210
150  J=J+1
160  GOTO 80
170  FOR I=1 TO N
180  PRINT A[I]
190  NEXT I
200  GOTO 230
210  I=I+1
220  GOTO 120
230  END


RUN


-8
-2
 0
 5
 6
 8

DONE
```

Exercise 5 — Identifying Increasing and Decreasing Functions

Micro Flow Chart
Exercise 5(a)

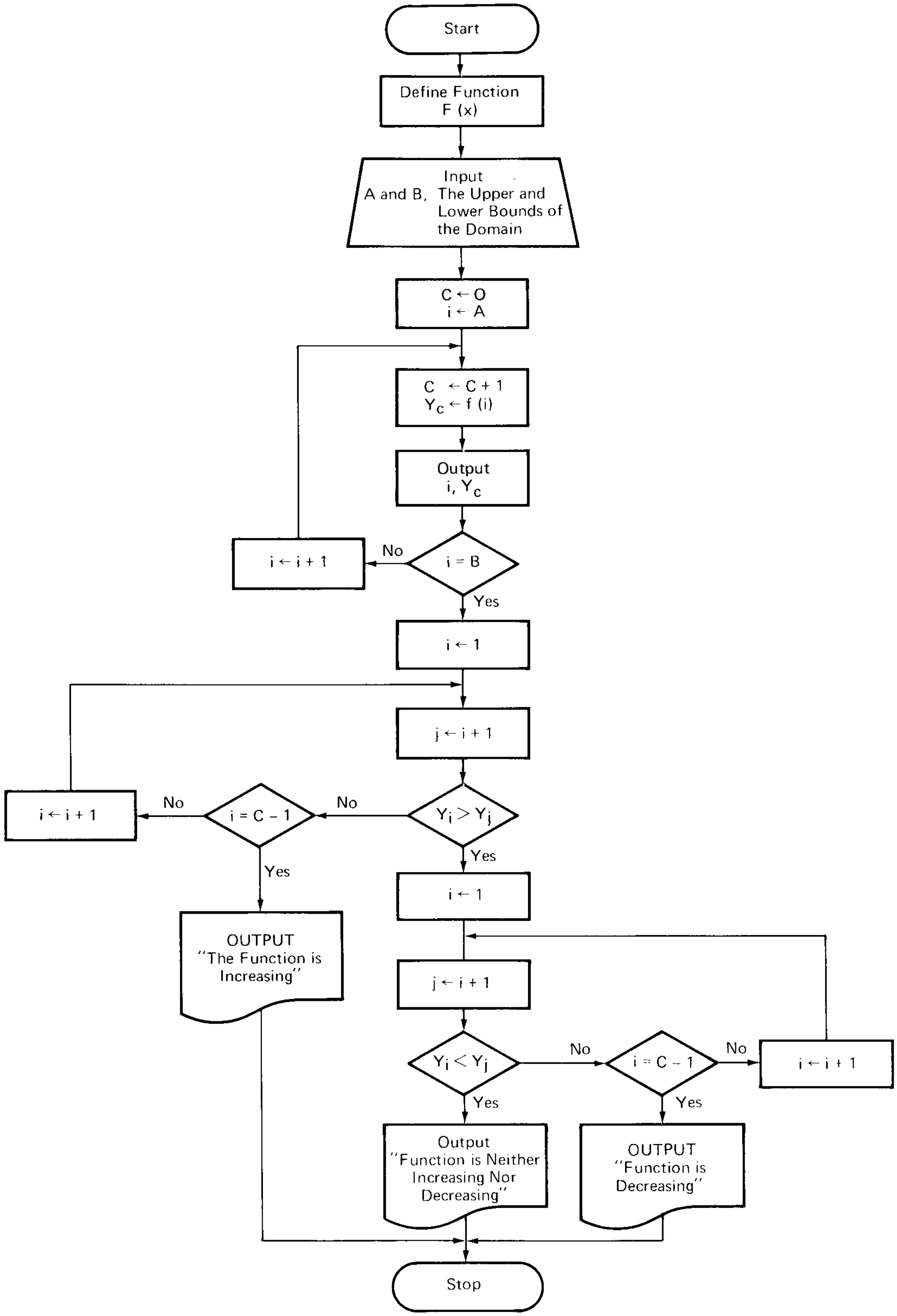


Example Program
Exercise 5(a)

```
10  DATA 7,.5,0,1.75,2,-1,-7,0,-4,-.5,-7,-3,-8,3,15
20  DATA 6,.5,-1,-1,0,7,-22,-7,6,3.5,-6,-5,0
30  DATA 6,-6,7,-5,2,-4,0,-3,2,0,0,3,1
40  REM--A PROGRAM TO DETERMINE IF A FUNCTION IS "INCREASING",
50  REM--"DECREASING" OR NEITHER. ON THE DATA LINE ENTER THE
60  REM--NUMBER OF ORDERED PAIRS IN THE SET FOLLOWED BY THE
65  REM--ORDERED PAIRS OF THE SET.
70  READ N
80  FOR I=1 TO N
90  READ X[I],Y[I]
100  NEXT I
110  LET I=1
120  LET J=I+1
130  IF X[I] <= X[J] THEN 220
140  LET D[I]=X[I]
150  LET R[I]=Y[I]
160  LET X[I]=X[J]
170  LET Y[I]=Y[J]
180  LET X[J]=D[I]
190  LET Y[J]=R[I]
200  IF I=N-1 THEN 250
210  GOTO 110
220  IF J=N THEN 370
230  LET J=J+1
240  GOTO 130
250  FOR I=1 TO N
260  PRINT X[I],Y[I]
270  NEXT I
280  I=1
290  J=I+1
300  IF Y[I] <= Y[J] THEN 400
310  I=1
320  J=I+1
330  IF Y[I] >= Y[J] THEN 430
340  PRINT "FUNCTION IS NEITHER  INCREASING OR DECREASING"
350  PRINT
360  GOTO 70
370  IF I=N-1 THEN 250
380  I=I+1
390  GOTO 120
400  IF I=N-1 THEN 460
410  I=I+1
420  GOTO 290
430  IF I=N-1 THEN 490
440  I=I+1
450  GOTO 320
460  PRINT "FUNCTION IS INCREASING"
470  PRINT
480  GOTO 70
490  PRINT "FUNCTION IS DECREASING"
500  PRINT
510  GOTO 70
520  END
```

```
RUN


-3                -8
-1                -7
-.5               -7
 0                -4
 .5                0
 1.75              2
 3                 15
FUNCTION IS INCREASING

-7                 6
-5                 0
-1                 0
 .5               -1
 3.5              -6
 7                -22
FUNCTION IS DECREASING

-6                 7
-5                 2
-4                 0
-3                 2
 0                 0
 3                 1
FUNCTION IS NEITHER  INCREASING OR DECREASING


OUT OF DATA  IN LINE 70
```

Micro Flow Chart
Exercise 5(b)



Example Program
Exercise 5(b)

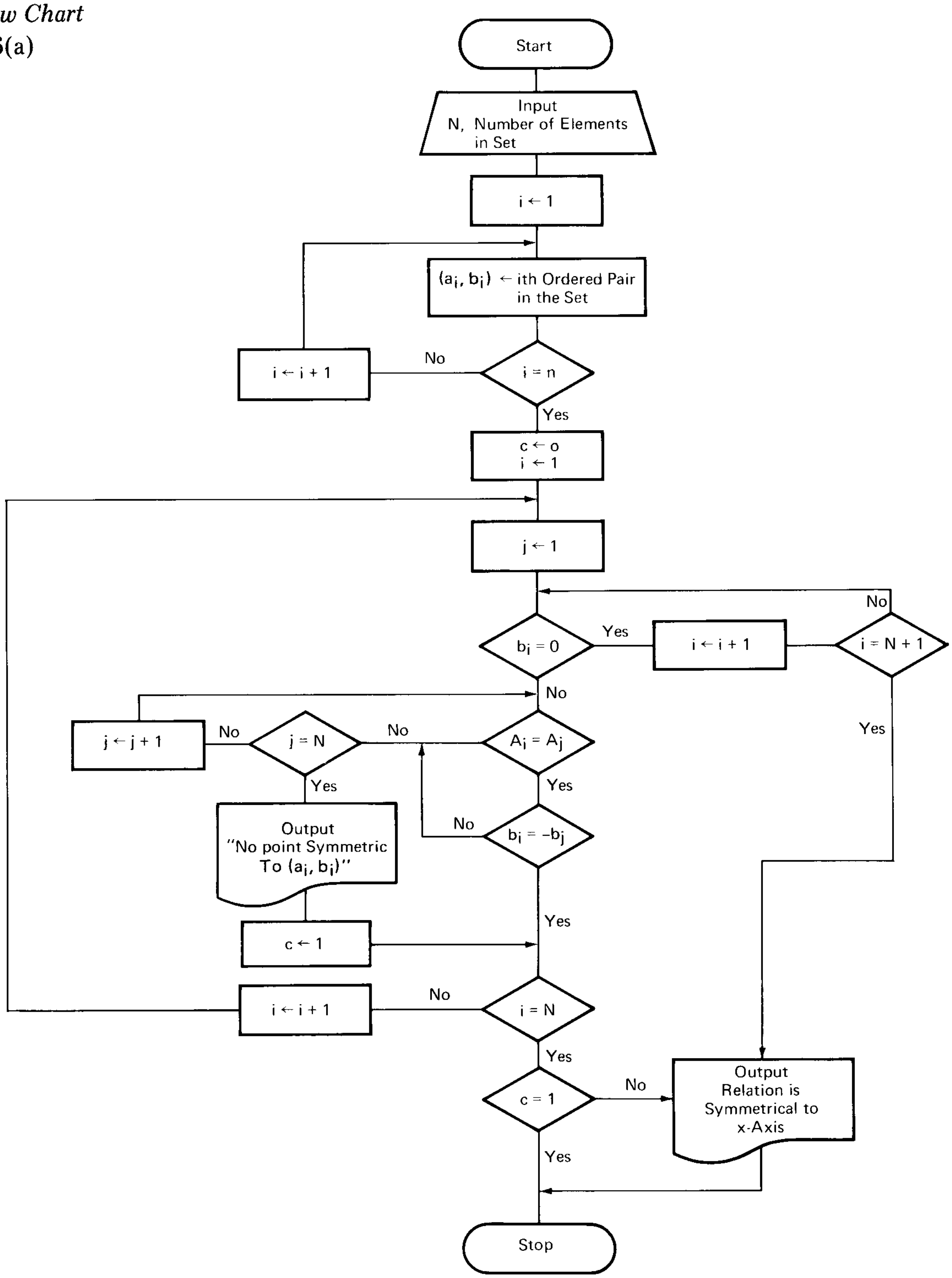
```
10  REM--A PROGRAM TO DETERMINE IF A FUNCTION IS "INCREASING",
20  REM--"DECREASING", OR NEITHER. ON LINE 60
30  REM--DEFINE THE FUNCTION.  INPUT UPPER AND LOWER DOMAIN.
35  C=0
40  DIM X[20],Y[20]
50  INPUT A,B
60  DEF FNP(X)=X↑2-X
70  FOR I=A TO B
80  C=C+1
90  Y[C]=FNP(I)
95  PRINT I;Y[C]
100  NEXT I
110  FOR I=1 TO C-1
120  J=I+1
130  IF Y[I]>Y[J] THEN 180
140  NEXT I
150  PRINT "THE FUNCTION IS INCREASING"
160  PRINT
170  GOTO 270
180  FOR I=1 TO C-1
190  J=I+1
200  IF Y[I]<Y[J] THEN 250
210  NEXT I
220  PRINT "THE FUNCTION IS DECREASING"
230  PRINT
240  GOTO 270
250  PRINT "THE FUNCTION IS NEITHER INCREASING OF DECREASING"
260  PRINT
270  END
```

```
RUN


?-3,3
-3      12
-2      6
-1      2
 0      0
 1      0
 2      2
 3      6
THE FUNCTION IS NEITHER INCREASING OF DECREASING


DONE
RUN


?0,5
 0      0
 1      0
 2      2
 3      6
 4      12
 5      20
THE FUNCTION IS INCREASING


DONE
60 DEF FNP(X)=6-X*2
RUN


?-5,5
-5      16
-4      14
-3      12
-2      10
-1      8
 0      6
 1      4
 2      2
 3      0
 4     -2
 5     -4
THE FUNCTION IS DECREASING


DONE
```

Exercise 6 — Symmetry

In this program, make sure that the students provide for the points (0,b), (a,0) (0,0) to map into themselves for x-axis, y-axis, and origin symmetry, respectively.

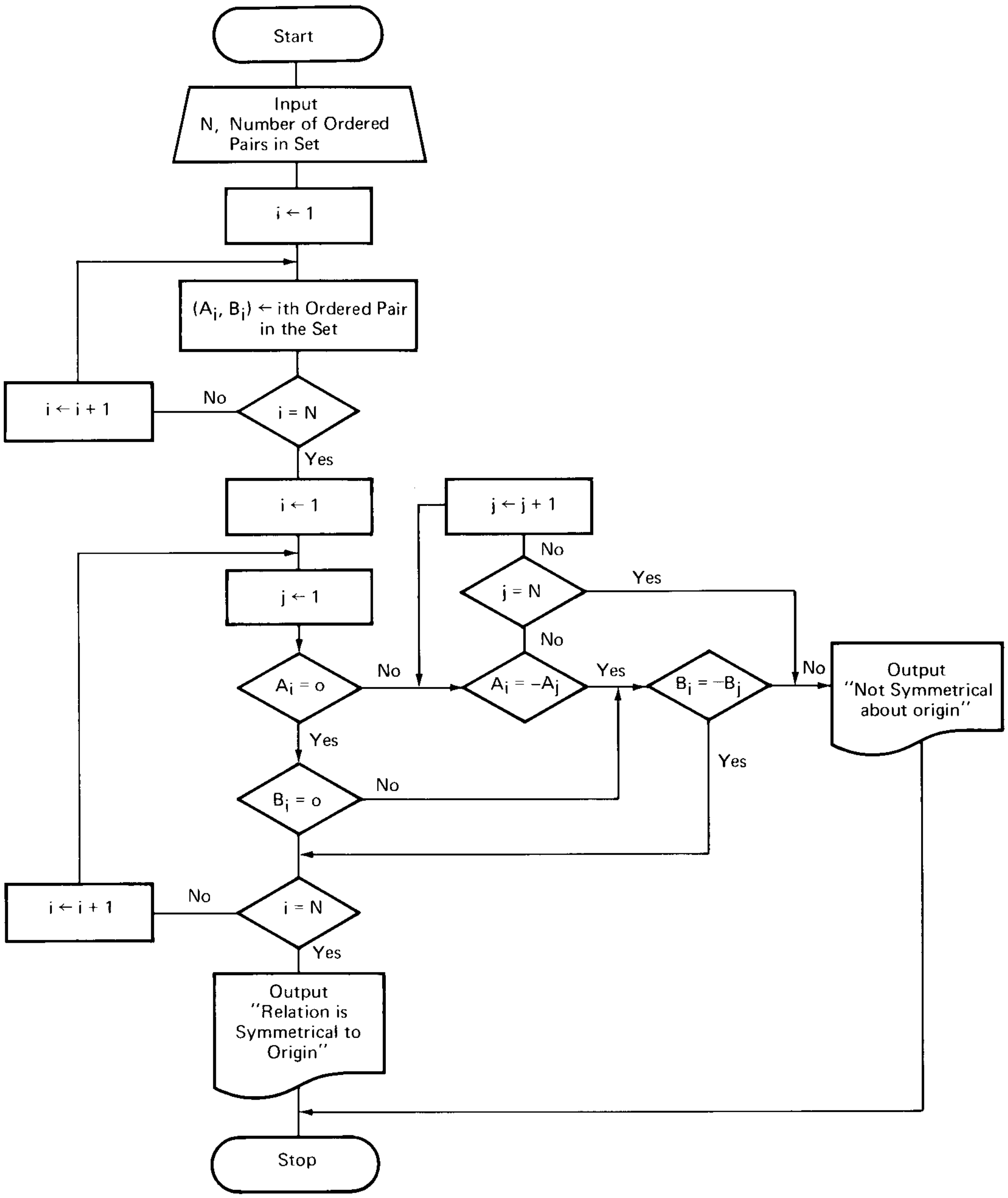
Micro Flow Chart
Exercise 6(a)



Example Program
Exercise 6(a)

```
10  REM--A PROGRAM TO MODEL THE DEFINITION OF SYMMETRY ABOUT
20  REM--THE X-AXIS.  ENTER ON THE DATA LINE N, THE NUMBER OF
30  REM--ORDERED PAIRS IN THE SET  AND THE ELEMENTS OF THE
40  REM--ORDERED PAIRS, PAIR BY PAIR.
50  DATA 9,6,3,-2,-1,-2,3,.7,-5,-2,1,.7,5,6,-3,-3,-2,-2,-3
60  DATA 7,2,15,7,0,6,-1,.5,-2,6,1,2,-15,.5,2
70  DATA 9,-5,1,7,3,6,.3,0,0,0,-2,6,-.3,7,-3,-5,-1,0,2
80  READ N
90  FOR I=1 TO N
100  READ A[I],B[I]
110  NEXT I
120  LET C=0
130  LET I=1
140  LET J=1
150  IF B[I]=0 THEN 200
160  IF A[I]=A[J] THEN 230
170  IF J=N THEN 250
180  LET J=J+1
190  GOTO 160
200  LET I=I+1
210  IF I=N+1 THEN 330
220  GOTO 150
230  IF B[I]=-B[J] THEN 290
240  GOTO 170
250  PRINT "NO POINT SYMMETRIC TO (";A[I];",";B[I];")"
270  LET C=1
290  IF I=N THEN 320
300  LET I=I+1
310  GOTO 140
320  IF C=1 THEN 340
330  PRINT "RELATION IS SYMMETRICAL ABOUT THE X-AXIS"
340  GOTO 80
350  END


RUN


NO POINT SYMMETRIC TO (-3     ,-2     )
RELATION IS SYMMETRICAL ABOUT THE X-AXIS
RELATION IS SYMMETRICAL ABOUT THE X-AXIS

OUT OF DATA  IN LINE 80
```

Micro Flow Chart
Exercise 6(b)



Example Program

Exercise 6(b)

```
10  REM--THIS PROGRAM MODELS THE DEFINITION OF SYMMETRY ABOUT
11  REM--THE ORIGIN.  ENTER, ON THE DATA LINE N, THE NUMBER OF
12  REM --THE ORDERED PAIRS AND THE ORDERED PAIRS, PAIR BY PAIR.
20  DATA 7,-7,1,-6,2,-5,3,-4,4,5,-3,6,-2,7,-1
30  READ N
40  FOR I=1 TO N
50  READ A[I],B[I]
60  NEXT I
70  LET I=1
80  LET J=1
90  IF A[I]=0 THEN 140
100  IF A[I]=-A[J] THEN 150
110  IF J=N THEN 160
120  LET J=J+1
130  GOTO 100
140  IF B[I]=0 THEN 200
150  IF B[I]=-B[J] THEN 200
160  PRINT "RELATION NOT SYMMETRICAL ABOUT THE ORIGIN"
163  GOTO 250
200  IF I=N THEN 240
210  LET I=I+1
220  GOTO 80
240  PRINT "RELATION IS SYMMETRICAL ABOUT ORIGIN"
250  END

RUN


RELATION NOT SYMMETRICAL ABOUT THE ORIGIN

DONE
20 DATA 8,-7,1,-6,2,-5,3,-4,4,7,-1,6,2,5,-3,4,-4
RUN


RELATION NOT SYMMETRICAL ABOUT THE ORIGIN

DONE
20 DATA 9,-7,1,-6,2,-5,3,-4,4,7,-1,6,-2,5,-3,4,-4,0,0
RUN


RELATION IS SYMMETRICAL ABOUT ORIGIN

DONE
20 DATA 9,-7,1,-6,2,-5,3,0,4,3,1,7,-1,6,-2,5,-3,-3,-1
RUN


RELATION NOT SYMMETRICAL ABOUT THE ORIGIN

DONE
```

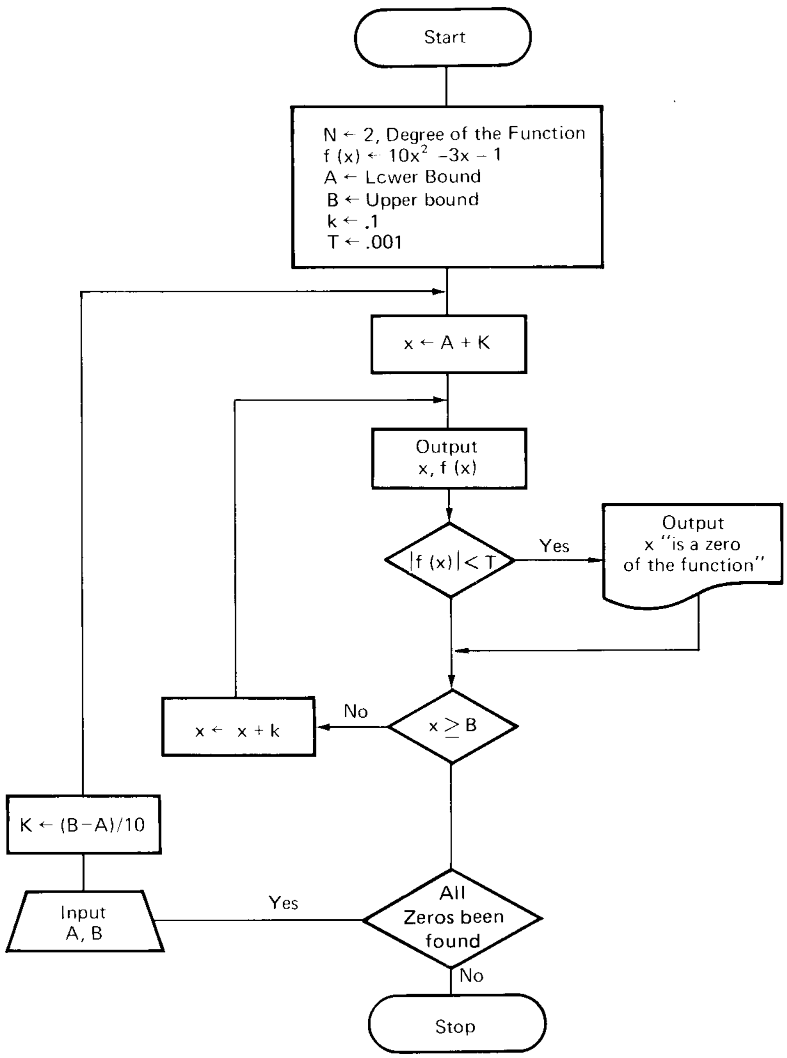
USING INFINITE FUNCTIONS TO INVESTIGATE OTHER FUNCTION PROPERTIES

Exercise 7 — The Zeros of a Function

This exercise is not intended to present a reliable method for finding zeros of all polynomials. Its main purpose is to convey the concept of a zero. The unreliability of this method does provide opportunity for discussion of computer error and problems associated with "not well-behaved" curves.

The following program provides a method to input new upper and lower bounds of an interval if a search for a given increment does not determine all zeros.

Micro Flow Chart
Exercise 7



Example Program
Exercise 7

```
10  REM--A PROGRAM TO DETERMINE THE ZEROS OF A POLYNOMIAL.
20  REM--ON LINE 40 ENTER THE POLYNOMIAL.
40  DEF FNF(X)=X↑3+3*X-1
50  PRINT "ENTER LOWER BOUND AND UPPER BOUND OF INTERVAL BEING SEARCHED."
60  PRINT "ALSO, THE TOLERANCE DESIRED FOR APPROXIMATE ZEROS."
70  INPUT A,B,T
80  K=.1
90  X=A+K
100  PRINT X,FNF(X)
110  IF ABS(FNF(X))<T THEN 150
120  IF X >= B THEN 190
130  X=X+K
140  GOTO 100
150  PRINT ""
160  PRINT X;"IS A ZERO"
170  PRINT ""
180  GOTO 120
190  PRINT "IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'"
200  PRINT "EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND"
210  PRINT "OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION"
220  PRINT "VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'."
240  INPUT Y
250  IF Y=1 THEN 290
260  INPUT A,B
270  K=(B-A)/10
280  GOTO 90
290  END
```

```
40 DEF FNF(X)=10*X↑2-3*X-1
RUN


ENTER LOWER BOUND AND UPPER BOUND OF INTERVAL BEING SEARCHED.
ALSO, THE TOLERANCE DESIRED FOR APPROXIMATE ZEROS.
?-1,1,.001
-.9              9.8
-.8              7.8
-.7              6.
-.6              4.4
-.5              3.
-.4              1.8
-.3              .8
-.2              2.38419E-07



-.2           IS A ZERO



-.1              -.6
-2.98023E-08     -1.
 .1              -1.2
 .2              -1.2
 .3              -1.
 .4              -.6
 .5               0



 .5           IS A ZERO



 .6               .8
 .7              1.8
 .8              3.
 .9              4.4
 1               6
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?1

DONE
```

```
40 DEF FNF(X)=X↑3+3*X-1


RUN


ENTER LOWER BOUND AND UPPER BOUND OF INTERVAL BEING SEARCHED.
ALSO, THE TOLERANCE DESIRED FOR APPROXIMATE ZEROS.
?0,1,.001
 .1             -.699
 .2             -.392
 .3             -.073
 .4              .264
 .5              .625
 .6              1.016
 .7              1.443
 .8              1.912
 .9              2.429
 1               3
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?0
?.3,.4
 .31            -4.02089E-02
 .32            -7.23195E-03
 .33             2.59371E-02
 .34             .059304
 .35             .092875
 .36             .126656
 .37             .160653
 .38             .194872
 .39             .229319
 .4              .264
 .41             .298921
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?0
?.32,.33
 .321           -3.92389E-03
 .322           -6.13809E-04


 .322        IS A ZERO
```

```
 .323              2.69818E-03
 .324              6.01220E-03
 .325              9.32789E-03
 .326              1.26457E-02
 .327              1.59655E-02
 .328              1.92873E-02
 .329              2.26109E-02
 .33               2.59366E-02
 .331              2.92642E-02
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?1

DONE



40 DEF FNF(X)=12*X↑2-7*X+1


RUN


ENTER LOWER BOUND AND UPPER BOUND OF INTERVAL BEING SEARCHED.
ALSO, THE TOLERANCE DESIRED FOR APPROXIMATE ZEROS.
?-1,1,.001
-.9                17.02
-.8                14.28
-.7                11.78
-.6                9.52
-.5                7.5
-.4                5.72
-.3                4.18
-.2                2.88
-.1                1.82
-2.98023E-08       1.
 .1                .42
 .2                8.00002E-02
 .3               -2.00002E-02
 .4                .12
 .5                .5
 .6                1.12
 .7                1.98
 .8                3.08
 .9                4.42
 1                 6
```

```
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?0
?.2,.3
 .21            5.92002E-02
 .22            4.08001E-02
 .23            2.48001E-02
 .24            1.12002E-02
 .25            1.19209E-07



 .25         IS A ZERO



 .26           -8.79979E-03
 .27           -1.51999E-02
 .28           -1.91998E-02
 .29           -2.07999E-02
 .3            -2.00002E-02
 .31           -1.67999E-02
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?0
?.3,.4
 .31           -1.67999E-02
 .32           -.0112
 .33           -3.20005E-03
 .34            .0072
 .35            .02
 .36            3.52001E-02
 .37            5.27997E-02
 .38            7.27999E-02
 .39            9.51998E-02
 .4             .12
 .41            .1472
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?0
?.33,.34
 .331          -2.26808E-03
 .332          -1.31202E-03
 .333          -3.32355E-04



 .333        IS A ZERO
```

```
 .334           6.72102E-04


 .334       IS A ZERO


 .335           1.70016E-03
 .336           2.75207E-03
 .337           3.82805E-03
 .338           4.92787E-03
 .339           6.05178E-03
 .34            7.19976E-03
 .341           8.37159E-03
IF ALL ZEROS HAVE NOT BEEN FOUND, TYPE '0'
EXAMINE THE ABOVE OUTPUT,ENTER THE LOWER AND UPPER BOUND
OF THE INTERVAL IN WHICH A SIGN CHANGE IN THE FUNCTION
VALUES OCCURS.  IF ALL ZEROS HAVE BEEN FOUND, TYPE '1'.
?1

DONE
```

Exercise 8 — Discontinuous Functions

(a), (b), (c). Since this exercise really requires only finding the zeros of a polonomial function, we will not write a separate program for this problem. The program written for Exercise 7 can be used with the following change:

line 160 PRINT "THE FUNCTION IS DISCONTINUOUS AT X = "; X

(d) The equations $f(x) = \frac{x-1}{(x-1)(x+3)}$ and $g(x) = \frac{1}{x+3}$ do not define the same function because $f(x)$ is not defined at $x = 1$ while $g(x)$ is.

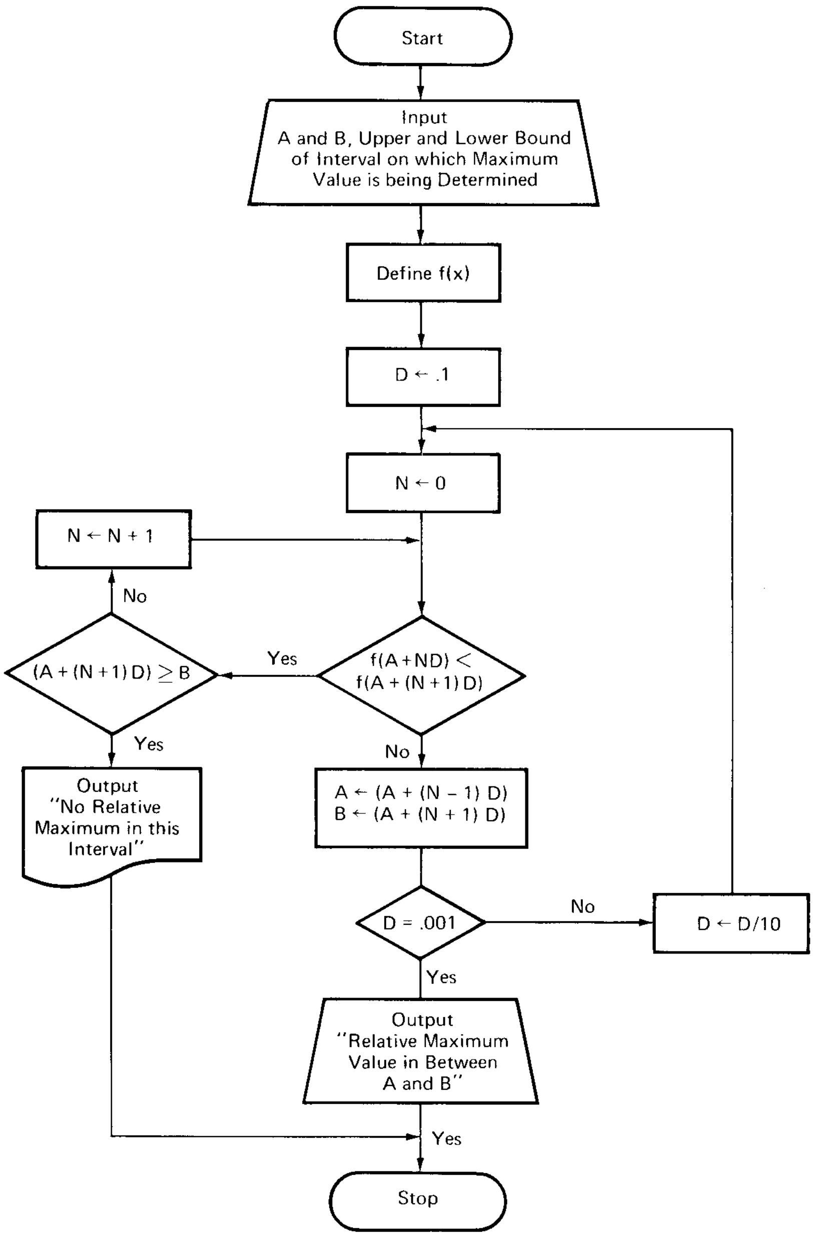
Exercise 9 — Relative Maximum Value

(a) You will notice that the technique used to approximate a relative maximum on an interval is not adequate unless the curve is "well-behaved" on that interval. Nevertheless, it is presented here with a well-behaved example since it quite effectively illustrates the concept of a relative maximum.

(b) This problem illustrates that relative maximum values do not exist for all intervals on a function. Students probably did not provide for this condition in the program for part (a), so they will need to modify their programs. The example program listed for part (a) does provide for intervals on which no relative maximums exist.

(c) This practical application of the program of part (a) should be of interest to the students.

Micro Flow Chart
Exercise 9(a)



Example Program
Exercise 9(a)

```
10  REM--A PROGRAM TO DETERMINE THE RELATIVE MAXIMUM VALUE OF
11  REM--A POLYNOMIAL FUNCTION F.  ON THE DATA LINE ENTER THE
12  REM--INTERVAL BOUNDS A AND B ON WHICH THE MAXIMUM VALUE IS
13  REM--TO BE DETERMINED.
20  DEF FNP(X)=X↑3-2*X↑2-X+2
30  READ A,B
40  DATA -1,1
50  LET D=.1
60  LET N=0
70  IF FNP(A+N*D)<FNP(A+(N+1)*D) THEN 130
80  LET A=A+(N-1)*D
90  LET B=A+(N+1)*D
100  IF D=.001 THEN 160
110  LET D=D/10
120  GOTO 60
130  IF (A+(N+1)*D) >= B THEN 180
140  LET N=N+1
150  GOTO 70
160  PRINT "RELATIVE MAXIMUM IS APPROXIMATELY ";FNP(A+N*D)
170  GOTO 190
180  PRINT "NO RELATIVE MAXIMUM ON THE GIVEN INTERVAL"
190  END

RUN

RELATIVE MAXIMUM IS APPROXIMATELY  2.11208

DONE



20 DEF FNP(X)=-4*X↑2+7*X+2
40 DATA 0,2
RUN

RELATIVE MAXIMUM IS APPROXIMATELY  5.06172

DONE
```

Example Program

Exercise 9(c)

```
10   REM--A PROGRAM TO DETERMINE THE RELATIVE MAXIMUM VALUE OF
11   REM--A POLYNOMIAL FUNCTION F.  ON THE DATA LINE ENTER THE
12   REM--INTERVAL BOUNDS A AND B ON WHICH THE MAXIMUM VALUE IS
13   REM--TO BE DETERMINED.
20   DEF FNP(X)=4*X↑3-42*X↑2+108*X
30   READ A,B
40   DATA 1,2
50   LET D=.1
60   LET N=0
70   IF FNP(A+N*D)<FNP(A+(N+1)*D) THEN 130
80   LET A=A+(N-1)*D
90   LET B=A+(N+1)*D
100   IF D=.001 THEN 160
110   LET D=D/10
120   GOTO 60
130   IF (A+(N+1)*D) >= B THEN 180
140   LET N=N+1
150   GOTO 70
160   PRINT "THE MAXIMUM VOLUME IS APPROXIMATELY";FNP(A+N*D);
165   PRINT "WHEN THE DEMINSION OF THE SQUARES CUT OUT IS BETWEEN";
166   PRINT A;"AND";B;"INCHES"
170   GOTO 190
180   PRINT "NO RELATIVE MAXIMUM ON THE GIVEN INTERVAL"
190   END
```

```
RUN


THE MAXIMUM VOLUME IS APPROXIMATELY 81.8714
WHEN THE DEMINSION OF THE SQUARES CUT OUT IS BETWEEN 1.696      AND
 1.704      INCHES

DONE
```